

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Patrik Kubiš

Bakalářská práce

Vedoucí práce: Ing. Svatopluk Štolfa, Ph.D.

Ostrava, 2021

Abstrakt

Tato bakalářská práce je písemnou zprávou popisující mé působení na individuální odborné praxi ve společnosti Tieto Czech s.r.o na pozici Capacity Analyst. Úvod práce obsahuje stručné představení společnosti, popis mého pracovního zařazení a hlavně popis projektu, na kterém jsem se v rámci své praxe podílel. U projektu se v práci zaměřuji na popis přípravy, výběr technologií, na podrobnější popis dílčích úkolů obsahující řešení úkolů, a to návrh i následnou implementaci. V závěru práce se zabývám zhodnocením projektu nasazeného v produkčním prostředí, získané znalosti a zkušenosti v průběhu praxe.

Klíčová slova

Python, SQL, RPA, Docker, Grafana

Abstract

This Bachelor's thesis is a written report describing my work on individual professional practice in the company Tieto Czech s.r.o in the position of Capacity Analyst. The introduction contains a brief description of the company, a description of my job and especially the project in which I participated in my practice. For the project, I focus on the description of preparation, selection of technologies. Detailed description of partial tasks, together with their solution, related to the design and subsequent implementation. In the end, I deal with the evaluation of the project deployed in the production environment. Gained knowledge and experience during practice.

Keywords

Python, SQL, RPA, Docker, Grafana

Poděkování

Rád bych na tomto místě poděkoval Tieto Czech s.r.o a Ing. Michalu Gelnarovi, Senior Project Manager, Innovatio Office, za umožnění absolvování individuální odborné praxe.

Petru Hruzíkovi, Senior DeVOPs Architect, Hi Ops RPA, za vedení a odborné rady v průběhu praxe.

A samozřejmě svému vedoucímu práce Ing. Svatopluku Štolfovi, Ph.D. za odbornou pomoc a konzultaci při vytváření této bakalářské práce.

Obsah

Seznam použitých symbolů a zkratek	6
Seznam obrázků	7
Seznam tabulek	8
1 Úvod	9
2 O firmě TietoEVRY	10
2.1 Popis společnosti a odborného zaměření	10
2.2 Pracovní zařazení	10
3 Analýza problému	12
3.1 Základní princip původního systému	12
3.2 Návrh principu nového systému	12
4 Použité technologie	14
4.1 Docker	14
4.2 Portainer	15
4.3 MariaDB	15
4.4 phpMyAdmin	15
4.5 Grafana	16
4.6 UiPath a RPA	16
4.7 Sharepoint	17
4.8 Microsoft Power Automate	17
5 Tvorba systému	18
5.1 Analýza původního systému	18
5.2 Nastavení virtuálního počítače	21
5.3 Tvorba databáze	22
5.4 Python skripty	23

5.5	Power Automate toky	25
5.6	RPA procesy	26
5.7	Grafana	29
6	Závěr	31
6.1	Zhodnocení systémů	31
6.2	Časová náročnost	33
	Literatura	34

Seznam použitých zkratek a symbolů

SQL	– Structured Query Language
T-SQL	– Transact-Structured Query Language
RPA	– Robotic process automation
ITIL	– Information Technology Infrastructure Library
NAS	– Network Attached Storage
SAN	– Storage Area Network
NBU	– NetBackup
VPN	– Virtual private network
JSON	– JavaScript Object Notation
IDE	– Integrated Development Environment
LTS	– Long term support

Seznam obrázků

3.1	Základní princip systémů	13
4.1	Rozdíl virtualizací	15
4.2	UiPath produktová architektura	17
5.1	Architektura serveru NetBackup Enterprise (třívrstvá)	19
5.2	Srovnání NAS a SAN	20
5.3	Ukázka rozhraní Portainer	21
5.4	Základní schéma NBU databáze	22
5.5	Ukázka Power Automate toku	25
5.6	Ukázka RPA sequence pro stahování reportů	26
5.7	Propojenost systému	28
5.8	Ukázka jednoho z NBU dashboardu v Grafaně	29

Seznam tabulek

6.1	Tabulka časové náročnosti ve dnech	33
-----	--	----

Kapitola 1

Úvod

Cílem této bakalářské práce je popsání mého působení ve společnosti Tieto Czech s.r.o. na pozici Capacity Analyst. Ve společnosti pracuji již od poloviny roku 2018, a proto jsem se rozhodl zde vykonat i individuální odbornou praxi, při které se projeví a doplní mé znalosti získané studiem vysoké školy v praktickém prostředí soukromé společnosti.

V této bakalářské práci uvedu společnost, která mi umožnila tuto praxi získat, a popíšu činnosti prováděné na projektu systému správy fyzické kapacity pro vybraná cloudová řešení (produkty) společnosti Tieto s.r.o v uplynulých dvou semestrech. V závěru budu hodnotit vytvořený systém a zkušenosti nabyté při cestě k jeho vytvoření a také využití znalosti z vysoké školy.

Kapitola 2

O firmě TietoEVERY

2.1 Popis společnosti a odborného zaměření

Společnost Tieto Czech s.r.o. je součástí nadnárodní nově vzniklé společnosti TietoEvry a jedná se o jednu z předních společností v oblasti digitálních služeb a softwaru poskytující kompletní IT řešení v mnoha různých odvětvích, mezi něž patří bankovníctví, veřejná správa, výrobní průmysl, lesnictví, pojišťovnictví, energetický průmysl atd. Celosvětově společnost aktivně působí ve více než 20 zemích a zaměstnává více než 24000 zaměstnanců.

V České republice společnost byla zapsána do obchodního rejstříku v roce 1995 a hlavní sídlem se zde od roku 2012 stalo Tieto Towers v Ostravě. Pobočka aktuálně zaměstnává okolo 2600 lidí po celé České republice díky možnosti vzdáleného pracoviště i téměř full time home office (nejen s ohledem na aktuální pandemickou situaci). Za zmínku stojí i nedávný vznik společnosti fúzí existujících společností Tieto a EVERY v roce 2019.

2.2 Pracovní zařazení

Ve společnosti pracuji již delší dobu. Nastupoval jsem těsně před začátkem studia na vysoké škole a zdálo se mi vhodné spojit své teoretické znalosti získávané na vysoké škole s praxí. Zároveň se mi zamlouvala skutečnost, že po dokončení studia již budu mít získané nějaké pracovní zkušenosti pro další kariérní růst.

Název mé pracovní pozice je Capacity Analyst a mým úkolem v Capacity Management je správa fyzické kapacity u sledovaných cloudových řešení (produktů), příprava analýz, doporučení na rozšíření, nákup či přesun hardwaru. Zároveň je důležité zajistit přehledné sjednocené zobrazení aktuálního stavu využití fyzické kapacity pro ostatní týmy a ITIL Practice. Nedílnou součástí mé práce je tedy efektivní získávání, uchovávání, zpracování a zobrazení dat.

Capacity Management je jedna z ITIL Practice. Je zodpovědná za správné dimenzování zdrojů IT, aby vyhovovaly současným i budoucím potřebám. Je to také jedna z pěti oblastí poskytování služeb ITIL. Efektivní správa kapacity je proaktivní, nikoli reaktivní. Dobrá správa kapacit zajistí, aby obchodní a servisní potřeby byly splněny s minimem IT zdrojů. [1]

Kapitola 3

Analýza problému

Z důvodu změny pracovní pozice mého Capacity Analyst kolegy jsem dostal pod správu navíc ke svým produktům a oblastem i jeho produkty a oblasti. Z toho třemi následujícími produkty (NBU, NAS a SAN) se budu zabývat v této bakalářské práci. Původní systém správy těchto produktů byl velmi časově náročný a obsahoval množství lidské práce. Objevil se tedy problém, jak co nejlépe zefektivnit práci a sledování dříve jmenovaných produktů pro ušetření času a lidské práce. Řešením tohoto problému má být nový systém popsáný v této bakalářské práci. V následujících pododstavcích popisují a následně i porovnávám princip původního systému a návrh nového systému.

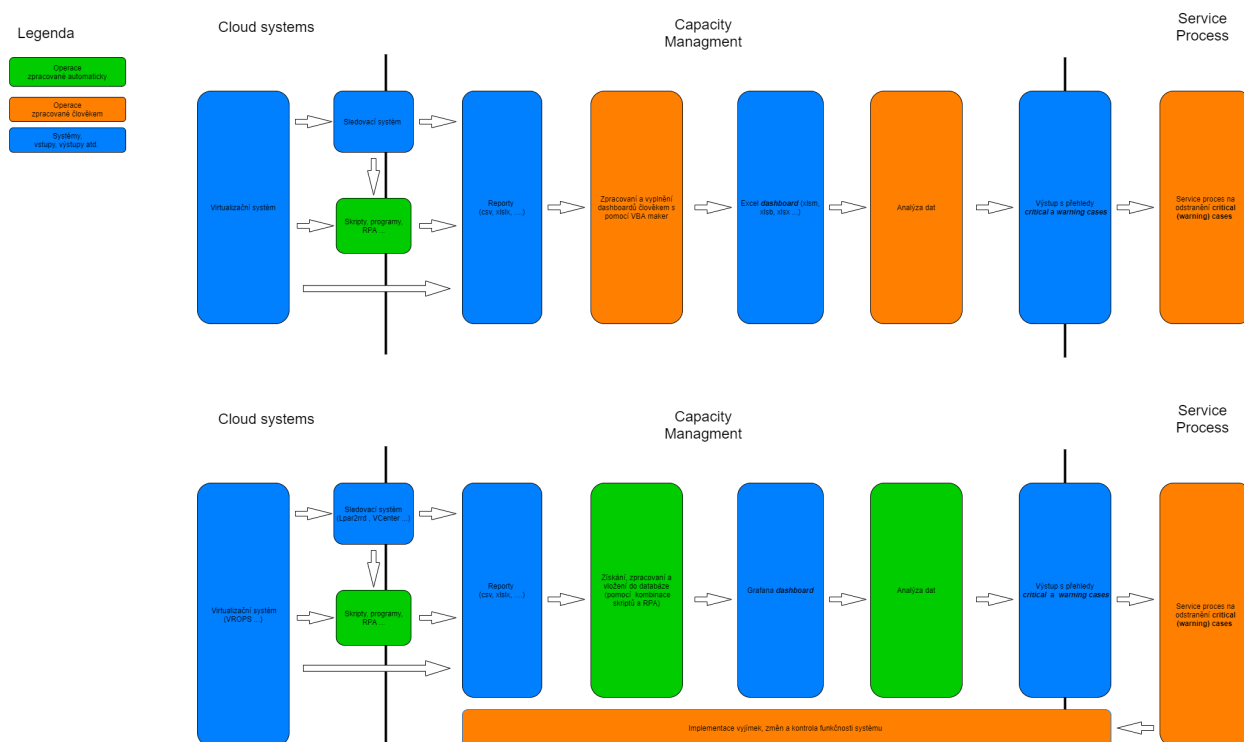
3.1 Základní princip původního systému

Původní proces sledování fyzické kapacity je založen na reportech a datech získávaných z mnoha zdrojů a předávaných přes sdílený mailbox. Data jsou následně člověkem kopírována do excel dashboardů a pomocí VBA maker zpracována. V posledním kroku jsou data opět kontrolována člověkem. Původní systém je třeba změnit i z důvodu nedostatečné kapacity Excel dashboardů a snižující se přehlednosti. Nahrazením bude systém obsahující databázi a zobrazení v Grafaně.

3.2 Návrh principu nového systému

Nový systém poběží na virtuálním počítači s operačním systémem Linux. V systému bude nainstalován Docker a v něm poběží kontejnery pro databázi (MariaDB), pro zobrazení (Grafana) a správcovské systémy (phpMyAdmin a Portainer).

Data budou získávána stejným způsobem jako původně, ze sdíleného mailboxu a to z důvodu časově náročného prověřování a získání přístupu k virtualizačním systémům a sledovacím nástrojům, které by nemusely být ani schválené. Jednodušší fungování a nakonec dostačující množství dat, které není třeba mít ani s vyšší časovou frekvencí, nám umožňuje tuto část systému zachovat.



Obrázek 3.1: Základní princip systémů

Vkládání dat bude umožněno RPA procesem běžícím v RPA UiPath orchestrátoru. Proces získá reporty a uloží je do interního úložiště robota, poté spustí Python skripty pro práci s daty a databázi (RPA umožňuje také práci s databázemi, ale z důvodu pomalejšího zpracování je použit Python skript, který je s ohledem na množství dat mnohem rychlejší). Zároveň proces bude kontrolovat nastavené prahové hodnoty a v případě překročení informuje uživatele z nastaveného seznamu zainteresovaných lidí (emailových adres, případně telefonních čísel nebo Microsoft Teams uživatelů). Data budou přístupná a zobrazená v Grafaně běžící v Dockeru, data se získají z databáze na základě nastavených SQL dotazů a T-SQL procedur. V systému bude obsažen i Microsoft Power Automate pro ukládání a stahování kontrolních dat na SharePoint.

Porovnání systémů je zobrazeno na obrázku, kde na první pohled vidíme úbytek lidské práce (oranžová barva) a nahrazení automatickým zpracováním (zelená barva).

Kapitola 4

Použité technologie

V této kapitole stručně popíši technologie použité v mém systému. Zároveň zde uvedu i některé výhody a důvody, proč jsem dané technologie zvolil.

4.1 Docker

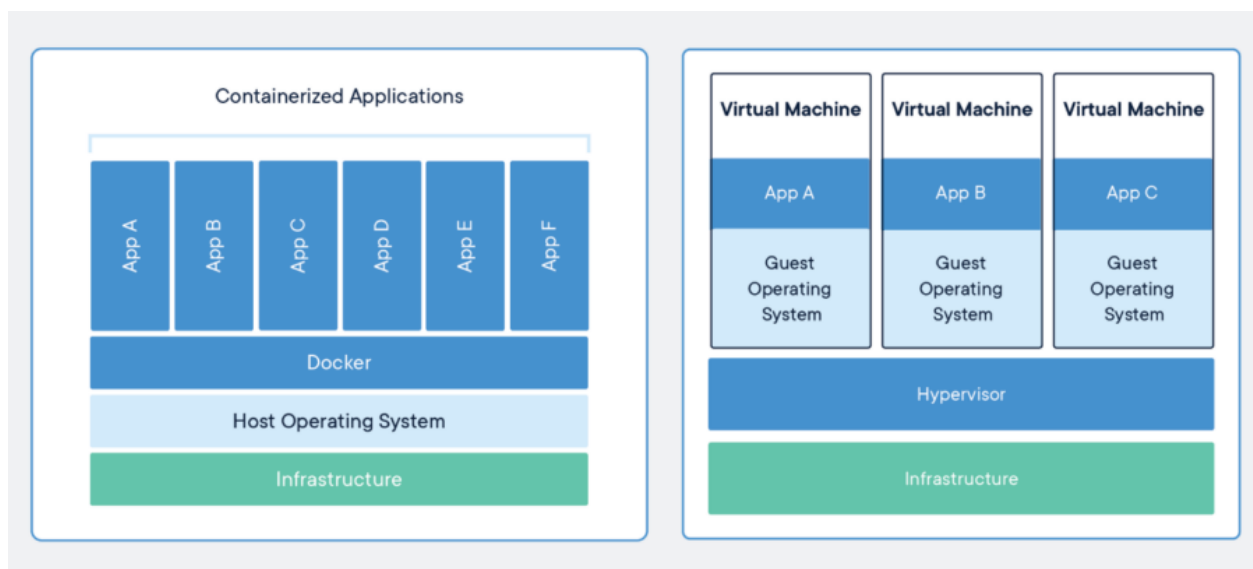
Docker je otevřený software, jehož cílem je poskytnout jednotné rozhraní pro izolaci aplikací do kontejnerů v prostředí macOS, Linuxu i Windows.

Kontejnerizace na rozdíl od plné virtualizace nepotřebuje mít operační systém pro každý virtuální server, jelikož všechny kontejnery běží v rámci jednoho operačního systému a sdílí mezi sebou hardware. Dalšími výhodami Dockeru je možnost izolování od prostředí a snadné použití v jiném prostředí.[2][3]

Základní komponenty Dockeru:

- Dockerfile - Jde o textový soubor s příkazy potřebnými pro tvorbu Docker Image. Popisuje prostředí pro kontejner a další komponenty kontejneru.
- Image - Jedná se o komprimovanou, samostatnou část softwaru bez lokálních úprav. Je potřebná k vytvoření a spuštění Docker kontejneru. Image mohou být použité ke snadnému přenosu kontejnerizovaných aplikací.
- Kontejner - Je běžící prostředí vytvořené z Image. Obsahuje tedy data nebo lokální změny.

Docker a i další software jsem volil hlavně proto, že se jedná o otevřený software, což je velmi důležité v komerčním prostředí, neboť představuje finanční úsporu. Dalším důvodem je snadná přenositelnost v případě změny virtuálního serveru.



Obrázek 4.1: Vlevo kontejnerová virtualizace, vpravo klasická plná virtualizace. Dostupné z [3]

4.2 Portainer

Portainer je otevřený platformový nástroj pro správu kontejnerových aplikací. Nástroj umožňuje snadnější správu kontejnerů, nastavení prostředí, tvorbu Image, sledování výkonu atd. Portainer jsem volil pro snadnější přístup k virtuálnímu serveru a lepší možnosti tvorby komponent Dockeru a i z důvodu velmi přehledného a jednoduchého grafického rozhraní. [4]

4.3 MariaDB

Je jedním z nejpopulárnějších relačních databázových systémů na světě. Vyvíjí ji původní vývojáři MySQL a je zaručeno, že zůstane otevřeným softwarem. Pro mou práci se hodil nejvíce tento druh databáze, jelikož je podporovaný téměř všemi otevřenými softwarovými aplikacemi. [5]

4.4 phpMyAdmin

Je opět otevřený software napsaný v jazyce PHP umožňující jednoduchou správu obsahu databáze MySQL prostřednictvím webového rozhraní. V mém systému slouží spíše jako kontrolní grafický nástroj pro databázi. Pro tvorbu databáze lze využít i jiné nástroje, které ale už nemusí být otevřeným softwarem, například DataGrip, MySQL Workbench, HeidiSQL atd. [6]

4.5 Grafana

Je multiplatformní otevřená webová aplikace pro analytiku a interaktivní vizualizaci.

Grafana má několik způsobů získání dat. Nejčastějším způsobem je získání pomocí SQL dotazů z připojené databáze. Zobrazovat data můžeme v Grafana dashboardech pomocí několika druhů vizualizačních panelů. Nejpoužívanějším vizualizačním panelem je Graf, Tabulka, Gauge panel. Celé nastavené dashbordy můžeme vkládat a zálohovat pomocí JSON souborů. [7]

4.6 UiPath a RPA

UiPath je globální softwarová společnost se sídlem v New Yorku, která vyvíjí platformu pro automatizaci robotických procesů (RPA). Základním IDE je UiPath Studio.

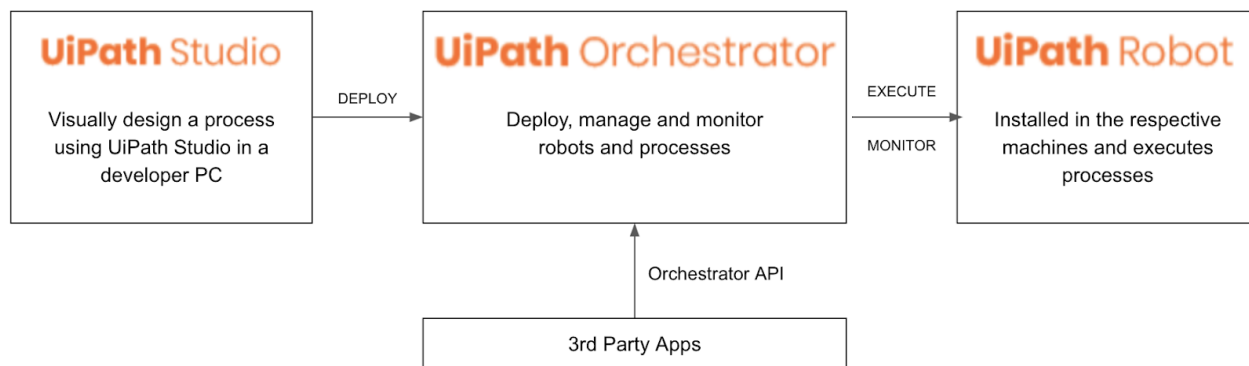
Robotic Process Automation je technologie, která umožňuje konfigurovat počítačový software neboli „softwarového robota“, který má napodobovat a integrovat akce člověka, pro jehož práci se využívají další systémy a software za účelem provedení procesu. Softwaroví roboti mohou využívat pro svou práci uživatelská rozhraní, ale zároveň i programové funkce a skripty, které by člověk bez dalšího softwaru využívat nemohl, například komunikace s databází, práce s daty. Samotný RPA proces je vytvářen pomocí UiPath aktivit, ty jsou nejčastěji vkládány do sekvencí nebo workflow.

Na rozdíl od jiných tradičních IT řešení umožňuje RPA organizacím automatizovat za zlomek dříve vynaložených nákladů a času. RPA je také možno aplikovat k využívání stávající infrastruktury, aniž by způsoboval narušení základních systémů, jejichž nahrazení by bylo obtížné a nákladné. [8]

RPA se nejčastěji používá k automatizaci procesů které:

- se velmi často opakují
- jsou náchylné k chybám
- zakládají se na pravidlech
- je potřeba je co nejrychleji automatizovat

Součástí UiPath platformy je UiPath Studio, které slouží pro tvorbu processů, UiPath orchestrátor pro správu a monitoring robotů a softwarový robot, ten může být attended, nebo unattended. Attended roboti se používají hlavně v testovací fázi, nebo u procesů vyžadujících částečnou interakci člověka.



Obrázek 4.2: UiPath produktová architektura. Dostupné z [9]

4.7 Sharepoint

SharePoint je webová platforma používaná hlavně ve firmách využívajících interní síť. Slouží primárně pro snadnější spolupráci v rámci týmu firmy nebo oddělení. Hlavní výhodou je možnost snadného sdílení souborů a dat.

Ve svém projektu využívám SharePoint primárně pro zálohu reportů a jejich případnou kontrolu a také pro ukládání nastavovacích souborů. [10]

4.8 Microsoft Power Automate

Jedná se o nástroj pro snadnou automatizaci procesů. Pro svou práci využívá toky, které jsou spouštěny vyvoláním akce. Velká databáze předdefinovaných, modifikovatelných procesů a zároveň možnost tvorby i komplexnějších procesů dělá z Microsoft Power Automate velmi oblíbenou administrativní automatizaci.

Příkladem toku může být uložení souboru na SharePoint, který je obsažen v emailu jako příloha s určitým předmětem nebo odesílatelem. [11]

Kapitola 5

Tvorba systému

Kapitola se zabývá samotnou tvorbou systému. V jednotlivých podkapitolách jsou postupně chronologicky seřazeny dílčí úkoly tak, jak jsem je postupně řešil a zpracovával. V první podkapitole jsou navíc stručně obecně popsány a vysvětleny sledované produkty.

5.1 Analýza původního systému

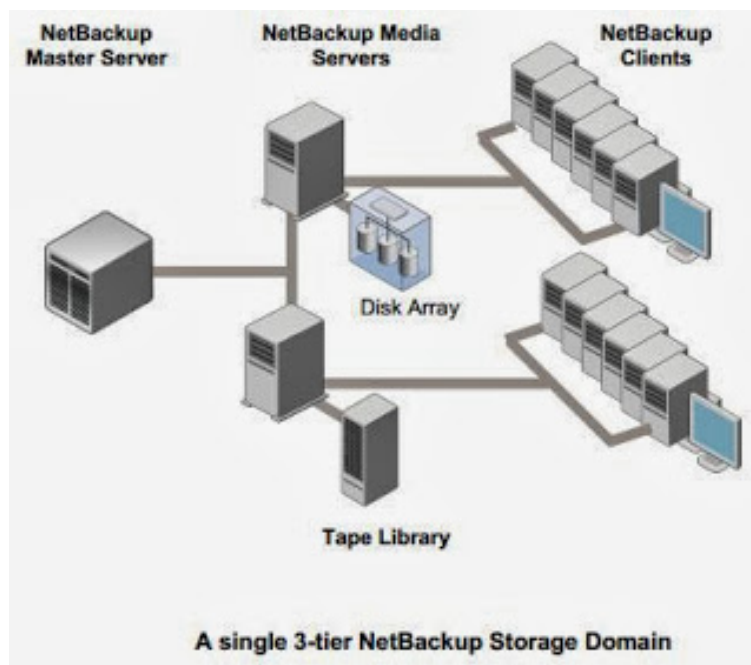
Původní systém běžel v Tietu bez větších problémů už delší dobu. Splňoval veškeré základní podmínky vyžadované od Capacity Managementu. Bohužel byl velmi náročný na lidskou práci a začínal být pomalu přeplněný, neboť kapacita Excelů je oproti databázi omezená a nedá se rozšířit.

Provedl jsem tedy celkovou obecnou analýzu původního systému a procesu pro sledování fyzické kapacity pro zjištění časové náročnosti. Mým dalším úkolem bylo získat základní teoretické znalosti principu fungování sledovaných produktů (NBU, NAS a SAN). Vyhledal jsem si tedy sledované produkty na internetu a pochopil jejich základní architekturu a fungování. Poté jsem se základními znalostmi jednotlivé produkty konzultoval s jejich architekty a produktovými manažery. Nakonec jsem důkladně analyzoval také původní dashboardy. Bylo nutno porozumět datovým kalkulacím, důležitosti a prioritizaci sledovaných a zobrazovaných dat, zjistit prahové hodnoty pro daná data, zjistit všechny výjimky a rizikové části a naučit se data zpracovávat. K této části jsem se často vracel a ověřoval si i správnost kalkulací. Zároveň jsem se vždy snažil pochopit, jak je vzorec sestaven. Případně jestli se nedá matematicky zlepšit a zpřesnit.

5.1.1 Architektura serveru NetBackup Enterprise (třívrstvá)

Doména úložiště NetBackup Enterprise Server se skládá z hlavních serverů (Master Server). Každý z nich má jeden nebo více serverů médií (Media Server) a více klientů NetBackup. Velcí zákazníci mají často několik domén NetBackup Storage rozložených v datových centrech. [12]

1. Master Server - Řídí veškeré aktivity v oblasti ochrany dat, od plánování a sledování záloh klientů po správu páskových médií a další. Kromě toho může mít Master Server NetBackup připojeno jedno, nebo více úložných zařízení pro zálohování lokálních dat, nebo klientských dat NetBackup v síti.
2. Media Server - Organizace s daty na několika místech, nebo s aplikacemi náročnými na data mohou distribuovat pracovní zátěž mezi servery nasazením serveru NetBackup Enterprise Server jako serveru médií. Master server je „mozek“ a Media Server jsou „pracovními koly“.
3. Diskpool/tape - Hardware, na kterém jsou uložena konkrétní data.



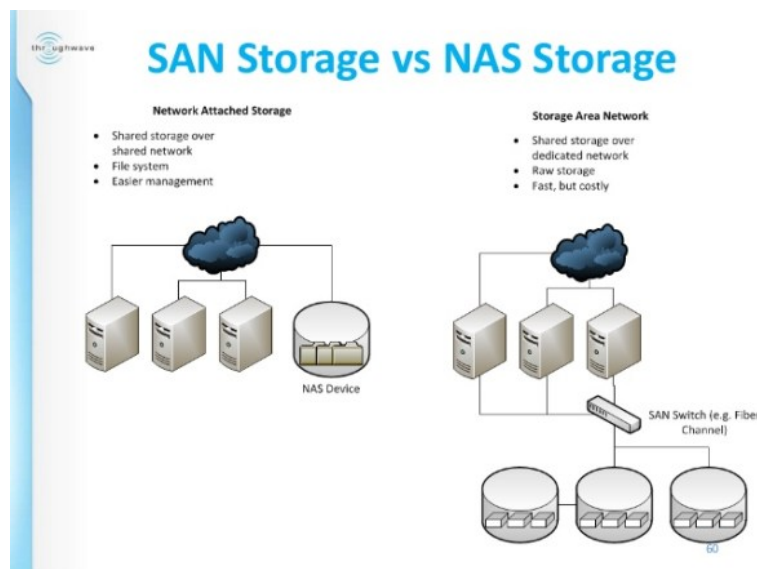
Obrázek 5.1: Architektura serveru NetBackup Enterprise (třívrstvá). Dostupné z [13].

5.1.2 Network Attached Storage

NAS je Storage technologie objevující se již od roku 1983. Jejím základem je co nejlepší datové úložiště k síti LAN. Kvalitní úložiště poskytuje dostatečnou ochranu a dostupnost dat. Clusterovaný NAS používá distribuovaného operačního systému, který běží současně na více serverech. Hlavním rozdílem mezi normálním a clusterovým NASem je schopnost distribuovat data a metadata mezi jednotlivými uzly, na které jsou zařízení napojena. To znamená, že jednotlivé NASy mohou poskytovat data jiných NAS, se kterými jsou spojeny. Podstatnou součástí profesionálních NAS bývá i nepřetržitý monitoring. Výhodou NAS oproti jiným technologiím jsou velmi nízké náklady a snadnější práce. [6] [14]

5.1.3 Storage Area Network

SAN představuje vysokorychlostní dedikovanou síť komunikující obvykle se serverem prostřednictvím optického kabelu. Vlastní síť je tvořena množstvím vzájemně napojených úložišť, tedy diskových polí a pásek. Výhodou je lepší propustnost dat a z důvodu většího počtu zařízení má také výkonnější zálohování a bezpečnost. SAN technologie je však relativně drahá. [14]

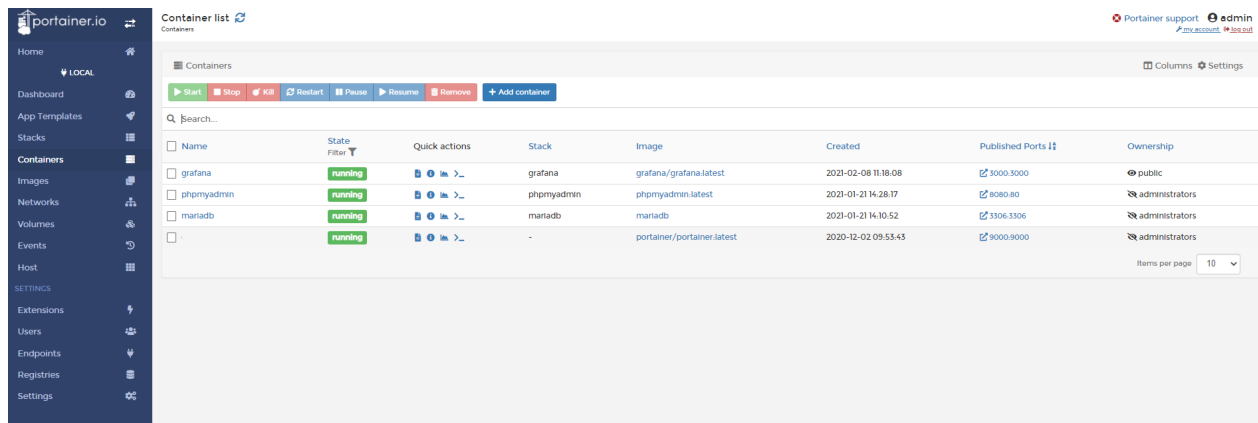


Obrázek 5.2: Srovnání NAS a SAN. Dostupné z [13].

5.2 Nastavení virtuálního počítače

Dalším krokem bylo nastavení virtuálního počítače, na kterém běží systém. Ten je dostupný pouze v rámci finské interní sítě. Přístup na něj je tedy umožněn pouze pomocí finské VPN, takže je zajištěna i částečná bezpečnost dat pouze pro interní uživatele. Na virtuálním počítači běží operační systém Linux Ubuntu 20.04.1 LTS. A v něm je nainstalován Docker. Pro snadnější správu v Dockeru běží kontejner Portainer, který umožňuje sledování kontejnerů a tvorbu jednotlivých Image, kontejnerů atd. Dalším kontejnerem je databáze MariaDB a nástroj pro správu databáze phpMyAdmin. Posledním kontejnerem je Grafana, která umožňuje zobrazování dat z databáze.

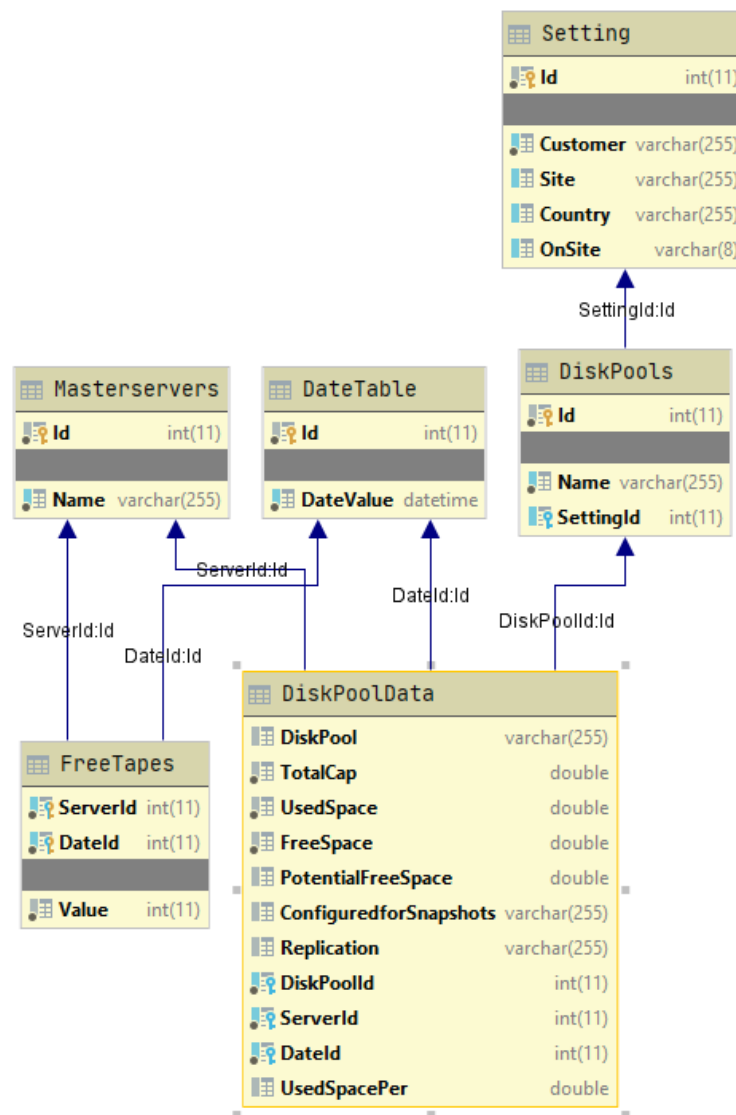
Na obrázku je ukázka zobrazení kontejnerů běžících na virtuálním počítači v Portaineru, ten umožňuje i snadný přístup k serveru, nebo třeba případnou úpravu kontejnerů a jejich sledování.



Obrázek 5.3: Ukázka rozhraní Portainer

5.3 Tvorba databáze

Pro systém bylo potřeba vytvořit tři oddělené MySQL databáze, pro každý sledovaný produkt jednu. U tvorby databází jsem tedy vycházel z původních tabulek a ze vstupních reportů. Databáze běží na virtuálním počítači v kontejneru MariaDB a jsou v nich uložena veškerá data. Malá část dat se dopočítává i přímo v databázi. Na obrázku je viditelné základní schéma databáze pro NBU.



Obrázek 5.4: Základní schéma NBU databáze

5.4 Python skripty

Pro každý sledovaný produkt jsem vytvořil několik Python skriptů. Z důvodu rozdílné struktury .csv reportů a účelu skriptů a také pro lepší přehlednost jsem nedával vše do jediného skriptu, navíc se některé skripty dají použít částečně i u jiných produktů. Skripty se dají rozdělit na dva základní druhy. Datové a nastavovací.

Datové skripty slouží ke zpracování a vložení dat do databáze. Načítají data ze souborů .csv a následně je nejčastěji pomocí knihovny Pandas zpracovávají v dataframech. Dotazují se do databáze a získávají ID, které přiřazují k datům, dopočítávají a upravují základní data. A nakonec vkládají data do příslušných tabulek v databázi. Pro ukázkou je zde uvedena funkce, která upravuje všechny .csv soubory ve složce tím, že jim smaže hlavičky, které překážejí při dalším zpracování. Nejdůležitější roli při samotném zpracování dat pomocí Pythonu je knihovna Pandas a modul MySQL Connector, které stručně popíši v následujících podkapitolách.

Nastavovací skripty pak slouží pro nastavení a rozklíčování hodnot. Příkladem pro objasnění může být změna názvu diskpoolu ve vstupním reportu, kdy nastavovací skript přejmenuje i zpětně diskpool v databázi tak, aby se nám nestalo, že ho v Grafaně uvidíme zobrazený jako dva diskpools, nebo se skript spouští v případě přidání nového diskpoolu. Tento skript se však nemusí spouštět pokaždé. Jeho spuštění nám zajistí RPA proces.

```
# Python funkce pro základní upravu csv souboru - smazání hlavičky
def dellfirstrows(base_directory,row):
for dir_path, dir_name_list, file_name_list in os.walk(base_directory):
    for file_name in file_name_list:
        if not file_name.endswith('.csv'):
            continue
        file_path = os.path.join(dir_path, file_name)
        with open(file_path, 'r') as ifile:
            line_list = ifile.readlines()
        with open(file_path, 'w') as ofile:
            ofile.writelines(line_list[row:])
```

Listing 5.1: Funkce pro základní úpravu .csv souboru - smazání hlavičky

5.4.1 Knihovna Pandas

Pandas je Python knihovna pro manipulaci a analýzu dat. Základním datovým typem knihovny je DataFrame, ten umožňuje základní práci s daty a jejich zobrazení. Pandas se nejčastěji používá pro zpracování .csv souborů. Ukázka zobrazuje jednoduché načtení dat z .csv souboru do DataFrame. [15]

```
# Python uložení dat z .csv do DataFrame
import pandas as pd
dataFin = pd.read_csv (base_directory+"inputFin.csv")
dfFin = pd.DataFrame(dataFin, columns= ['Master Server', 'Disk Pool', 'Disk Pool
    Type', 'Total Capacity(TB)', 'Used Space (TB)', 'Free Space (TB)', 'Potential
    Free Space (TB)', 'Configured for Snapshots', 'Replication'])
```

Listing 5.2: Uložení dat z .csv do DataFrame

5.4.2 MySQL Connector

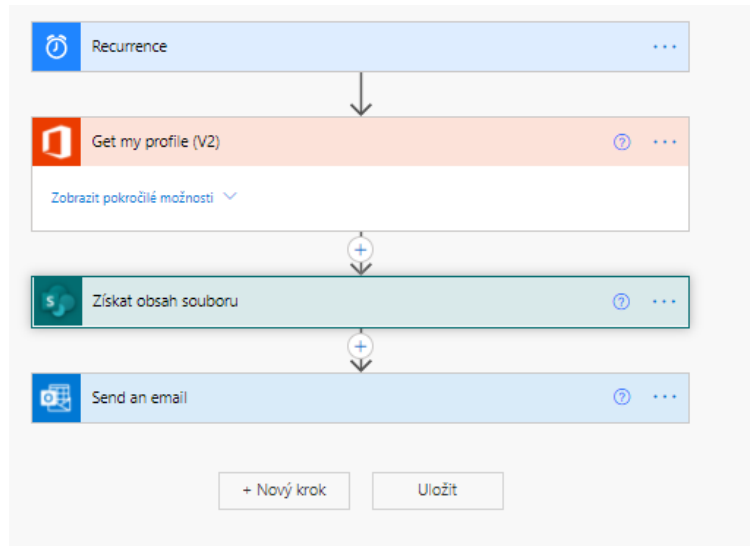
Python modul MySQL Connector slouží k propojení s databází a pro práci s ní pomocí SQL. Ukázka obsahuje funkci, která pomocí cursoru vkládá data na základě vloženého generovaného SQL dotazu a hodnot. Nejedná se o knihovnu, takže je potřeba před použitím tento modul nainstalovat. [16]

```
# Python funkce pro vkládání dat do databáze
def insert(sql, val):
    mycursor.execute(sql, val)
    mydb.commit()
```

Listing 5.3: Funkce pro vkládání dat do databáze

5.5 Power Automate toky

Pomocí toků je zajištěno kontrolní ukládání reportů, díky kterému si může kdokoli bez přístupů a znalostí databází zobrazit reporty, a dále také toky zajišťují stahování souborů obsahujících nastavovací data, jako je popis pro jednotlivé disky, nebo třeba seznam osob, které chtějí dostávat upozornění na Critical a Warning case. Tento proces je i zobrazen na obrázku. Power Automate umožňuje snadnou a rychlou práci se základními operacemi, a to hlavně v Office 365. Existuje zde velká řada i již předpřipravených toků, které se dají případně upravit.



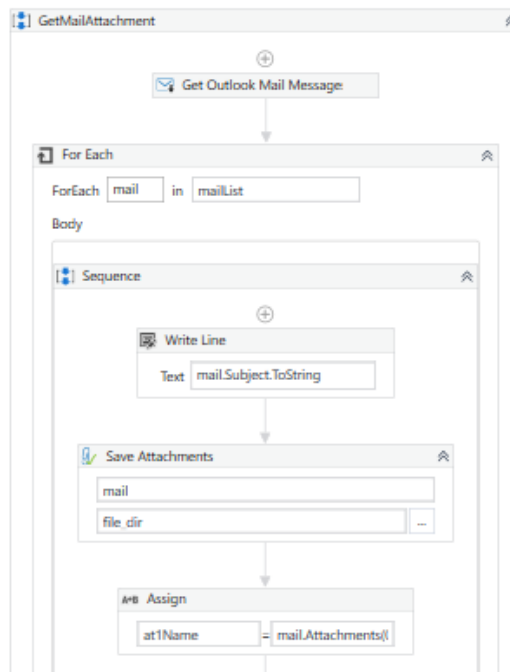
Obrázek 5.5: Ukázka Power Automate toku

Tok zobrazený na obrázku se spustí pokaždé v předem definovaný čas, přihlásí se k profilu, stáhne soubor ze sharepointu a pošle jej jako přílohu mailu.

5.6 RPA procesy

Pro systém jsem vytvářel opět tři samostatné RPA procesy, pro každý produkt jeden. Systém nemůže obsahovat pouze jeden proces, jelikož každý produkt se zpracovává ve specifický den a čas. Proto je nutné mít 3 oddělené procesy, které postupně spouští orchestrátor v naplánovaný čas a den. Procesy slouží jako jakési propojení a doplnění všech použitých technologií.

RPA proces zajišťuje získání vstupních souborů reportů z mailboxu a následně kontrolu správnosti souborů, zda nedošlo při jeho generování k chybě. Zajišťuje úpravy, jako je změna nastavení, názvu a případný převod souborů .xlsx na .csv (u produktu NAS a SAN). Soubory následně ukládá do interního úložiště softwarového robota, kde k nim přistupují Python skripty. Ty postupně spouští robot pomocí cmd aktivit (UiPath Studio nabízí i aktivitu na spuštění přímo Python skriptu. Bohužel je zatím podporována do Python verze 3.6 a pro zpracování dat potřebuji verzi 3.7.5 a vyšší). Toto se bude však v budoucnu určitě dát nahradit aktivitou pro Python po změně podporovaných verzí. Po skončení Python skriptů se vygeneruje seznam s Critical a Warning case. Zde už je použita aktivita UiPath pro přístup do databáze, jelikož se jedná zpravidla jen o malé procento dat, a navíc je zde i nízká složitost dotazu. Nakonec je seznam posílán pomocí aktivity e-mailem osobám obsaženým v nastavovacím souboru, který je předáván robotovi z Sharepointu pomocí Power Automate toku. Toto externí uložení nastavovacího souboru je hlavně z důvodu snadného přidání a odebrání osob. V případě SMS alertingu nebo Teams alertingu je posílán mail na interní bránu, která vygeneruje SMS nebo Teams zprávu.

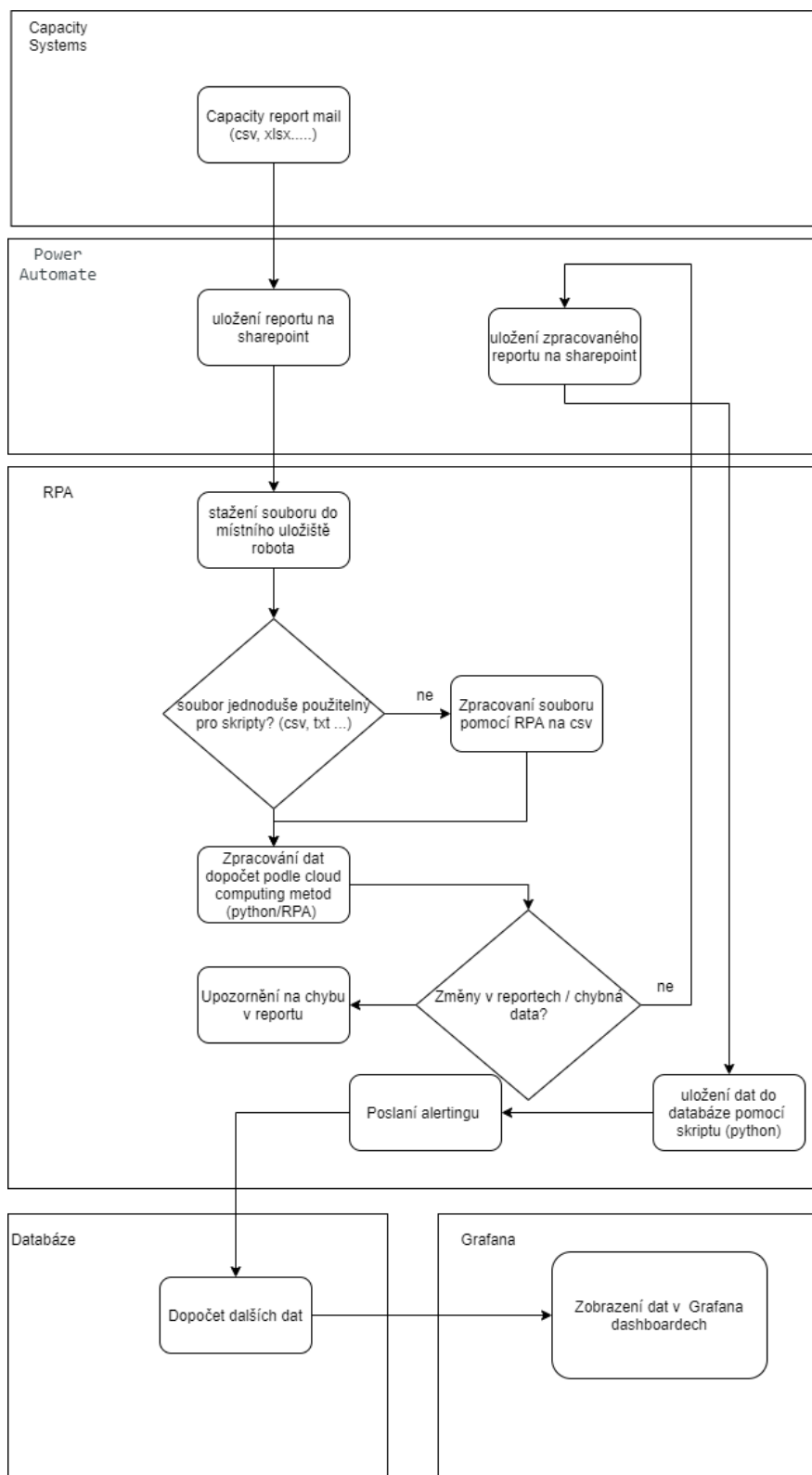


Obrázek 5.6: Ukázka RPA sequence pro stahování reportů

Celý RPA proces běží v UiPath orchestrátoru, kde je sledovaný, a v případě chyby posílá upozornění na chybu a informace o chybě včetně screenu.

Pro ukázkou je zde i zobrazena sequence, která stahuje reporty z generovaných mailů v mailboxu. Na začátku sequence je aktivita, která získá a uloží si do proměnné zprávy splňující podmínky (odesílatel, předmět a obsah). Následně v cyklu projde zprávy a uloží jejich přílohy do interního úložiště softwarového robota pomocí aktivity.

Celá propojenost všech technologií je zjednodušeně viditelná na schématu na další stránce.



Obrázek 5.7: Propojenost systému

5.7 Grafana

Posledním a časově nejnáročnějším je samotný grafický výstup. Ten je vytvořen v prostředí Grafana. Zde se v jednotlivých dashboardech zobrazují grafy, tabulky, Gauge panely atd. Pomocí SQL dotazů a T-SQL procedur zobrazuje data z databáze. Grafický vzhled je výsledkem konzultace s lidmi, pro které je dashboard určený a je možno ho případně doplňovat na základě jejich zpětné vazby.

V Grafaně je vytvořeno několik uživatelů a skupin, kteří mají nastavená práva pro jednotlivé složky (obsahující dashboardy, nejčastěji rozdělení podle produktů). Některé dashboardy jsou ještě rozděleny na více částí podle států a dá se v nich filtrovat na základě zadaných informací, například zakazníků nebo názvů. Za zmínku stojí i speciální sumarizační dashboardy, které slouží pro kvartální zhodnocení nárůstu objemu součtů všech zákazníků v jednotlivých oblastech, zde jsou většinou data získávána pomocí T-SQL procedur.



Obrázek 5.8: Ukázka jednoho z NBU dashboardu v Grafaně

Na obrázku vidíme v pravém horním rohu zobrazované časové rozpětí, které si můžeme sami zvolit. Pod ním jsou na pravé straně zobrazeny hodnoty z posledního reportu a vedle nich historický vývoj hodnot. V levém horním rohu jsou proměnné, pomocí kterých si můžeme data filtrovat. Pod nimi se nachází odkazy na související dashboardy pro daného přihlášeného uživatele. V prostřední části vidíme Gauge panel přehledně zobrazující criticalitu, po levé straně je popis a vysvětlivky pro tabulku zobrazenou ve spodní části obrázku. Ta je nejdůležitější pro detailnější zobrazení dat, obsahuje podrobnější informace o sledovaném produktu, trendy a další specifitější informace.

Pod tabulkou jsou zobrazeny ještě grafy zachycující vývoj detailnějších hodnot z tabulky, toto však už není na obrázku viditelné. Bývá tam častokrát zobrazena velikost daného diskpoolu v čase nebo vývoj jeho použitelné kapacity, trend využití a také proměnlivá predikce.

Kapitola 6

Závěr

Tato práce mě seznámila s celou řadou nových technologií, jako je kontejnerizace pomocí Dockeru, zobrazení dat Grafanou, a také rozšířila mé znalosti ve tvorbě RPA procesů a znalosti programovacího jazyka Python, a to hlavně v oblasti práce s daty. Při tvorbě jsem použil své znalosti získané na vysoké škole, a to hlavně analytickou matematiku, znalosti programování, z toho nejvíce pak jazyku Python, dále znalosti a zkušenosti s tvorbou MySQL databází, SQL a T-SQL.

6.1 Zhodnocení systémů

Zde se zaměřím na celkové zhodnocení obou systémů, původního i nového systému, a popíši zde jejich výhody a nevýhody.

6.1.1 Původní systém

Původní systém byl funkční a v počáteční fázi velmi účinný a také nezbytný. V případě sledování nového produktu je vždy potřeba vytvořit určité dočasné řešení, které je snadno upravitelné a umožní nám nastavit si prahové hodnoty. Také nás naučí rozpoznat nejdůležitější části a oblasti, kde vzniká nejvíce alertů a chyb. Hlavní výhodou je snadnější tvorba tohoto systému. Bohužel v pozdějších fázích již ztrácí systém přehlednost a má omezenou kapacitu. A hlavně vyžaduje velké množství lidské práce.

Tento systém, který byl ve společnosti již delší dobu zaběhnut je plně nahrazen mnou vytvořeným novým systémem.

6.1.2 Nový systém

Nový systém je plně funkční bez zásahu člověka. V případě přidání nových zákazníků v rámci produktu je systém plně automatizovaný a není třeba provádět žádné, ani částečné úpravy. Ty se provádějí pouze v případě nastavení specifických výjimek, tento případ se však už u nových zákazníků nestává. Výjimky jsou většinou u starších zákazníků, kteří už nebudou navyšovat kapacitu, a nehrozí u nich riziko nedostatku kapacity, proto mohou částečně porušovat nastavená pravidla pomocí výjimek. V případě nastavení nového produktu se využije stejný virtuální počítač a je možnost použít i částečně RPA proces a Python skripty, které se však musí modifikovat pro nový produkt. Databáze a Grafana dashboardy se musí vytvářet pro každý produkt specificky.

Výhody toho systému jsou tedy následující:

- Časová a finanční úspora. Po nasazení a nastavení systému už netřeba k funkci lidskou práci.
- Neomezené množství dat díky možnosti navyšování velikosti databáze.
- Libovolná frekvence reportingu.
- Časově nenáročné a téměř okamžité zpracování po příchodu reportů.
- Možnost dashboardy upravovat pro konkrétní osoby, týmy (individuálně nastavené dashboardy pro uživatele).
- Automatický alerting po překročení thresholdů.
- Automatické změny v dashboardu v případě přidání nových dat.

Nevýhodou může být pouze prvotní časově náročnější nastavení produktu, větší technická náročnost a znalost.

6.2 Časová náročnost

Celkem jsem na tvorbě systému strávil 54 pracovních dnů z toho 26 dnů na tvorbě NBU produktu, 14 dnů na NAS a stejný počet i na SAN produktu. Podrobněji lze vidět detailnější rozpis dnů strávených na řešení dílčích úkolů v tabulce níže.

Tabulka 6.1: Tabulka časové náročnosti ve dnech

	NBU	NAS	SAN
Analýza systému	2	2	2
Nastavení virtuálního počítače	2	-	-
Vytvoření databáze	1	1	1
Python skripty	7	4	4
RPA proces	6	2	2
Nastavení Grafany a tvorba dashboardů	8	5	5
Celkem pro jednotlivá řešení	26	14	14

Literatura

1. *Capacity Management: ITIL Version 2*. [B.r.]. Dostupné také z: <https://www.helpsystems.com/solutions/optimization/service-delivery-til-version-2/capacity-management>.
2. JEBAVÝ, Bc. Josef. *Jak začít správně používat Docker*. [B.r.]. Dostupné také z: <https://blog.josefjebavy.cz/unix/docker>.
3. GRYGAŘÍKOVÁ, MICHAELA. *Docker, Kubernetes a kontejnery. Jak fungují a proč je chtít*. [B.r.]. Dostupné také z: <https://www.master.cz/blog/docker-kubernetes-kontejnery-jak-funguji-proc-je-chtit/>.
4. PORTAINER.IO. *Open Source Container Management GUI for Kubernetes, Docker, Swarm*. [B.r.]. Dostupné také z: <https://www.portainer.io/>.
5. *MariaDB Foundation*. 2019-11. Dostupné také z: <https://mariadb.org/>.
6. *PhpMyAdmin*. Wikimedia Foundation, 2019-08. Dostupné také z: <https://cs.wikipedia.org/wiki/PhpMyAdmin>.
7. *Grafana*. Wikimedia Foundation, 2021-03. Dostupné také z: <https://en.wikipedia.org/wiki/Grafana>.
8. INC., UiPath. *What is Robotic Process Automation - RPA Software*. [B.r.]. Dostupné také z: <https://www.uipath.com/rpa/robotic-process-automation>.
9. *COMMUNITY*. [B.r.]. Dostupné také z: <https://dev.joget.org/community/display/KBv6/Integration+with+UiPath+Robotic+Process+Automation>.
10. UNIVERZITA, Masarykova. *Microsoft SharePoint*. [B.r.]. Dostupné také z: <https://it.muni.cz/sluzby/microsoft-sharepoint>.
11. MSFTMAN. *Začínáme s Power Automate - Power Automate*. [B.r.]. Dostupné také z: <https://docs.microsoft.com/cs-cz/power-automate/getting-started>.
12. SUDHAKARAN, Anup. [B.r.]. Dostupné také z: <http://serversignal.blogspot.com/2013/11/netbackup-architecture.html>.
13. 17, Martin January. *When To Use NAS And When To Use SAN*. 2020-01. Dostupné také z: <https://techsightings.com/when-to-use-nas-and-when-to-use-san/>.

14. ČERNÝ, Jiří. *NAS vs. SAN - jak na správu dat?* 2009-08. Dostupné také z: <https://www.svethardware.cz/nas-vs-san-jak-na-spravu-dat/27556>.
15. *Nauč se Python!* [B.r.]. Dostupné také z: <https://naucese.python.cz/lessons/intro/pandas/>.
16. [B.r.]. Dostupné také z: https://www.w3schools.com/python/python_mysql_getstarted.asp.